

University of Southern Mississippi

Gulf Park Campus

Rewards Shop Tracker Program

Development Documentation

Jackie Diep

CSC 414 G001

Professor Michaela Elliott

13 November 2020

## **Section 1.0: Introduction**

This program will belong to an employee rewards shop physically located in MadeUpMart. The rewards shop uses a point based system where points are allocated to outstanding employees annually. Each awarded employee is given 100 points that expire at the end of the year. All points must be spent at once, and any remaining points will be voided on checkout. Up to twenty items maximum may be purchased from the rewards shop at once. This program will function as a digital catalog within the store to help employees make selections among the store items.

## **Section 1.1: Purpose**

This document will detail the development process of the Rewards Shop Tracker Program commissioned by MadeUpMart's current owner, "Paul M. Talies", and any notable References shall be listed, as well as the manner of inspiration. A description of the Requirements initially specified and expanded upon throughout the month of November of the year 2020 by the commissioner and a record of future additions shall be documented and updated whenever necessary. A detailed explanation of how each requirement is addressed to both the current developer and commissioner's standards shall also be documented and maintained in the Design section. Testing methodologies and platforms will be satisfactorily detailed with documented results described in the Appendix of the document. Maintenance details and Versioning history will be found towards the end of the document, also containing the names of any software relevant to the program, and finally a User Manual for employees will be included afterwards.

## **Section 2.0: References**

- Takes inspiration from the Caesar's Total Return Rewards store that Harrah's Casino uses for outstanding employees
  - Uses point system like Caesar's
  - Uses money for up to an amount of remaining points like Caesar's (Point Overlap)
- Uses the Walmart online store for inventory options and point costs
- Takes inspiration from various arcade ticketing systems as a physical location that has a digital catalog with a few choice options among each type of reward

## **Section 3.0: Requirements**

- Project commissioned solely by and Requirements detailed in cooperation with MadeUpMart Owner "Paul M. Talies"

### **Initial Requirements**

1. The program shall display items in a menu format and accept numerical input for selection
  - a. The menu shall have 10 items
  - b. The menu shall have an exit condition
  - c. The menu shall respond and reprompt from an invalid response
  - d. The menu shall be easily alterable
2. The program shall always initialize the same amount of points
  - a. The program shall always start with 300 points

- b. The initial point amount shall be easily alterable
  - c. The program shall not allow more than this amount of points to be spent
- 3. The program shall have an output
  - a. The program shall output the items obtained
  - b. The program shall output the points spent
  - c. The program shall output the points voided

### Requirements Refinement

- 1) The program shall display items in a menu format and accept numerical input for selection
  - a) The menu shall display 10 item types
  - b) The menu shall display 3 options for each item type when selected by the user
    - i) The options shall be budget, mid-range, and luxury
  - ~~c) The menu shall have an exit condition~~
  - ~~d) The menu shall respond and reprompt from an invalid response~~
  - ~~e) The menu shall be easily alterable~~
- ~~2) The program shall always initialize the same amount of points~~
  - ~~a) Unchanged~~
- ~~3) The program shall have an output~~
  - ~~a) Unchanged~~

## Requirements Finalization

- ~~1. The program shall display items in a menu format and accept numerical input for selection~~
  - ~~a. *Unchanged*~~
- ~~2. The program shall always initialize the same amount of points~~
  - ~~a. *Unchanged*~~
- ~~3. The program shall have an output~~
  - ~~a. *Unchanged*~~
4. The program shall allow employees to pay in USD to make up for insufficient points.
  - a. The program shall allow users to convert USD for up to 50 points
  - b. The ratio shall be \$1.50 : 1 point
5. The program shall allow users to purchase up to a maximum quantity of items
  - a. The maximum quantity shall be 20

*\*\*\*Commissioner has requested that the next product release version shall allow users to enter a quantity to purchase more than one item simultaneously.*

## Section 4.0: Design

### Initial Design

- To Satisfy Requirement 1
  - Display the items using cout statements that print the item names
  - Accept input from the user using cin statements

- An array will be used to hold the cost of each of the ten items
- A switch-case structure will be used to allow the user's input to control program execution
  - Default case of the switch will be used to respond in case of an error in user input and re-prompt
  - '1' through '10' cases of the switch will be used to decide between inventory items
  - '0' case of the switch will be the exit condition of the menu
- Menu Item Names can easily be altered by changing print statements
- Menu Item Costs can easily be altered by changing cost array initialized values
- To Satisfy Requirement 2
  - Initialize an initial point variable equal to 300
  - Easily alterable by changing value of the initPoints variable in code
- To Satisfy Requirements 3
  - Two parallel arrays will be used that will hold the names and point costs of items purchased. These will be used to print a receipt at the end of the program of items purchased.
  - A variable will be used to track points spent.
    - This variable will be used to print the points spent
    - This variable will be subtracted from initial points to determine points voided
    - This variable will not be allowed to be greater than initial points

## Critical Design

- To Satisfy Requirements 1
  - Display the item types using cout statements that print the item types
  - ~~- Accept input from the user using cin statements~~
  - Two 2d parallel arrays will be used to hold the names and costs of the 3 items of each of the 10 item types
  - A switch-case structure will be used to allow the user's input to control program execution
    - '1' through '10' cases of the switch will be used to decide between item types
    - Menu Item Type Names can easily be altered by changing print statements
    - Costs of the individual items can be easily altered by changing the initialized array values
    - Names of the individual items can be easily altered by changing the initialized array values
    - ~~- Default case of the switch will be used to respond in case of an error in user input and re-prompt~~
    - ~~- '0' case of the switch will be the exit condition of the menu~~
- ~~- To Satisfy Requirement 2~~
- ~~- Unchanged~~
- ~~- To Satisfy Requirement 3~~

~~—Unchanged~~

## Final Design

~~—To Satisfy Requirement 1~~

~~—Unchanged~~

~~—To Satisfy Requirement 2~~

~~—Unchanged~~

~~—To Satisfy Requirement 3~~

~~—Unchanged~~

- To Satisfy Requirement 4

- A variable will be used to hold the conversion rate between USD and points
- Three variables will be used to handle conversion calculation
  - One variable will remember the amount of points overlapped (the points that must be converted to USD)
  - The second variable will be used to hold the product of the pointOverlap variable and the conversion rate
  - The final variable will calculate the final cost of the overlap including sales tax. This variable will be output by the program.

- To Satisfy Requirement 5

- A variable will be used to hold the maximum quantity. Currently this is equal to twenty.



- After a purchase, the quantity of items purchased will be compared to the maximum quantity. If greater than or equal to the maximum quantity, the menu will proceed to checkout.

## **Section 5.0: Test Plans**

- Basic testing occurred within the team office and on a variety of systems, including recently purchased and unconfigured machines, to ensure code is able to function without system specific requirements.
- Tested at the MadeUpMart Employee Rewards Store digital catalog terminal with a UI implemented by a separate team.

## **Critical Testing Plan**

After the developer has visually tested the proper appearance of the menu, the correct item displays, and the proper calculation of checkout with various combinations of purchases; the code will be tested at the store. Code will be tested at the MadeUpMart Employee Rewards Store terminal by stakeholders and commissioners. After an initial teaching period, stakeholders should be allowed to interact with the digital catalog at their leisure, and the developer should track the proper calculations and menus as they appear. Any suggestions should be noted, and if no critical errors appear, a receipt printer should be tested with the catalog to test integration. If properly outputted, give the code to the UI team for regression testing.

## **Final Testing Plan**

After the developer has visually tested the proper appearance of the menu, the correct item displays, and the proper calculation of checkout with various combinations of purchases; the code will be tested at the store. Code will be tested at the MadeUpMart Employee Rewards Store terminal by stakeholders and commissioners. The developer should guide users through each section of the menu and help simulate various orders to ensure that every newly implemented feature of the program is tested. In this version, users will be informed of the new Maximum Quantity feature or the USD Overlap Usage feature and requested to try to break it. If this process is satisfactory, and the program does not break, the program should be attached to a ticket printer that output is sent to. If output is proper and formatted, the final phase of the code is ready for regression testing with UI commissioned from the out of team source. Both teams should review the implementation of the UI, and if the tests are satisfactory, the code should be ready for release.

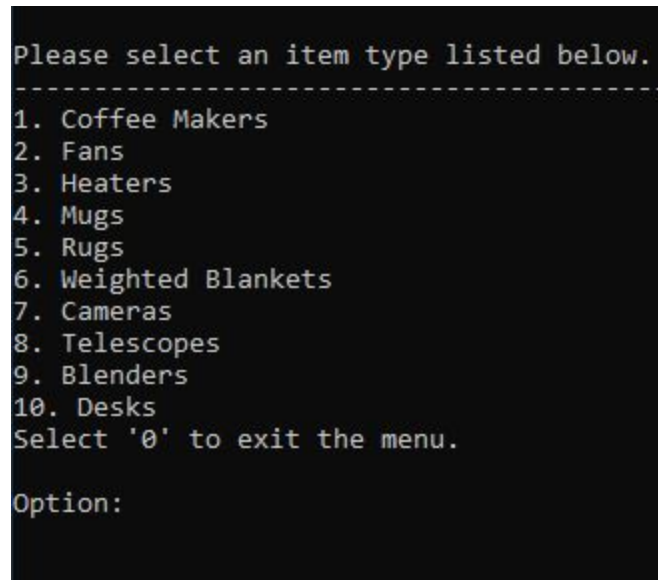
## Appendix

### Final Test Results

Program was finished in a short amount of time, and there were no critical problems in testing or implementation. A few calculation issues and infinite loop issues existed during the critical testing phase, but those were ironed out before the final testing phase. Testers were primarily the developer and the employees of the MadeUpMart store, though the commissioner “Paul M. Talies” did briefly review and simulate an order during final testing phase.

### Requirement 1:

- Initial Menu display is satisfactory
  - 10 items



```
Please select an item type listed below.
-----
1. Coffee Makers
2. Fans
3. Heaters
4. Mugs
5. Rugs
6. Weighted Blankets
7. Cameras
8. Telescopes
9. Blenders
10. Desks
Select '0' to exit the menu.
Option:
```

- Budget, Mid-Range, Luxury options

```

***Reward Store: Coffee Makers
1.      Budget: Mainstays 5 Cup Black Coffee Maker
        Cost: 9.88
2.      Mid-Ranget: Hamilton Beach FlexBrew 2-Way Coffee Maker
        Cost: 99
3.      Luxury: Keurig K-Duo Plus Coffee Maker
        Cost: 229.99
*****

```

- Exit condition

```

10. Desks
Select '0' to exit the menu.

Option: 0

Have a nice day!

```

- Invalid input handling

```

Invalid item type menu entry.
Please try again.

Please select an item type listed below.
-----
1. Coffee Makers
2. Fans
3. Heaters

```

- Menu is easily alterable due to array initialization section and a low amount of total items

## Requirement 2

- Initial point requirement is satisfied
  - Always initializes 300 points

```

Initial Points: 300

Please select an item type listed below.

```

- Initial points easily alterable at the variable initialization

### Requirement 3

- Output requirement is satisfied
  - Points spend, voided, and items purchased

```
*****
Number of Items Purchased: 5
Items Obtained:

Soleil Electric Personal Ceramic Mini Heater 250W Indoor
Point Cost: 10.94

Soleil Electric Personal Ceramic Mini Heater 250W Indoor
Point Cost: 10.94

Comfort Zone 9-inch 2-Speed Portable Box Fan
Point Cost: 11.63

Lasko 20-inch Win Tunnel 3-Speed Fan with Remote
Point Cost: 43.81

Cauldryn Coffee Smart Mug
Point Cost: 89.95

*****

Points spent: 167.27
Points voided: 132.73
Points over the limit: 0
Money spent: $0.00

Thank you for shopping at MadeUpMart's Employee Rewards Store
```

### Requirement 4

- USD to Point conversion due to Point Overlap feature implemented and satisfied
  - Up to 50 points from USD at \$1.50 : 1 point

```
*****
Points spent: 300
Points voided: 0
Points over the limit: 34.03
Money spent: $54.62
```

## Requirement 5

- Maximum purchase quantity implemented and satisfied
  - 20 Maximum quantity

```
Purchased item name: Mainstays 5 Cup Black Coffee Maker
Purchased item cost: 9.88

Maximum number of items purchased

Points remaining: 102.4

*****
Number of Items Purchased: 20
Items Obtained:
```